RE-ORDER NO. 64-123

67 f.

N64-19353
Code-1

NASA - CR-53738

March 1964

Final Report

A MATHEMATICAL STUDY OF IMAGE ANALYSIS

By: *A. R. Tobey, Staff Scientist*
Systems Engineering Laboratory
and
R. O. Duda, Research Engineer
Applied Physics Laboratory

Prepared for:

JET PROPULSION LABORATORY
4800 OAK GROVE DRIVE
PASADENA, CALIFORNIA

SUBCONTRACT 950687 UNDER NAS7-100

SRI Project No. 4621

This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration under Contract NAS7-100.

Copy No............

1 of 67

ABSTRACT

This report gives the results of an experimental study of the use of feature detection for television bandwidth compression. The goal is to determine from a set of patterns a set of simpler patterns, or features, so that each of the original patterns can be formed, at least approximately, by superposing the features. Four algorithms for determining features from patterns are described, and the results of experiments using these algorithms are compared and evaluated.

2

CONTENTS

I INTRODUCTION

The great technical problems involved in transmitting lunar and planetary television pictures to Earth call for a careful consideration of potentially useful bandwidth compression techniques. A method of coding television signals that reduces the number of bits needed to describe a picture allows either a reduction in the bandwidth required of the transmission channel or a more efficient use of an unalterable channel. This report gives the results of an investigation of the use of feature detection* for such purposes.

The pictures considered in this study are limited to black and white patterns defined on a finite grid or retina. Thus for a retina with N cells, N bits of information are sufficient to specify any pattern. If all of the 2^N possible patterns were equally probable, and if exact transmission of the patterns were required, no further reduction in the average number of bits needed would be possible.

The classical coding results[2,3,4] show that a reduction can be obtained if there are statistical constraints on the patterns. In this study very specific constraints on the patterns are assumed. Every pattern in the set of patterns to be transmitted is assumed to be composed (at least approximately) of some combination of K features. A <u>feature</u> is itself a pattern, usually different from those in the set to be transmitted, and features are combined by superposition. Clearly, if $K < N$, the number of bits needed to specify a pattern can be reduced from N to K.

---

*In this report the word "feature" is used in a precise sense that will be defined shortly. Feature detection was originally developed in connection with pattern recognition problems; a discussion of its role in that application is given in Ref. 1. (References are listed at the end of the report.)

In general, neither the features nor the number of features is known *a priori*. They must be determined from a statistically representative set of sample patterns, the training set. If features are found that can be combined to reconstitute (at least approximately) the patterns in the training set, it is reasonable to expect that they can also be used to reconstitute (approximately) those patterns of which the training set is representative. Then if the number of features is less than the number of retinal cells, bandwidth compression can be achieved at the cost of equipment complexity. The transmitter indicates which of the K features are to be used in reconstituting any pattern, and the receiver, which has previously been supplied with the features derived from the training set, is capable of reconstituting any pattern similar to those in the training set.

Most of the effort in this study was concentrated on finding and evaluating algorithms for determining the features from a training set. Because little theoretical guidance was available, the search for effective algorithms was almost entirely experimental, consisting of a large number of digital computer runs and little in the way of theoretical support. The evaluation of the proposed algorithms was also done experimentally. Questions such as the following were investigated:

(1) For fixed K, how closely can the patterns in the training set be approximated?

(2) How good are these features for reconstituting patterns in an independent testing set?

(3) How sensitive are the features to variations in parameters in the algorithm, the number of features allowed, etc.?

(4) How dependent is the performance of the algorithm on the nature of the patterns?

The results of this investigation are given in this report. First an algorithm is described that attempts to determine, for any pattern set, a set of features from which the patterns can be reconstituted with minimum error. The results of using this algorithm were reported previously,[5] and are briefly summarized here. A more detailed description is then given of other algorithms that were studied in the hope of finding an algorithm requiring less computation time, thereby allowing experiments with larger retinas. Results of experiments with these algorithms are given.

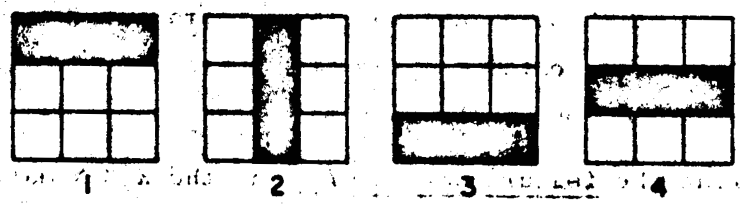
II THE EXHAUSTIVE ITERATIVE ALGORITHM

It is convenient to represent a black and white pattern on a retina with N cells by an N-tuple of ones and zeros. The representation can be achieved by scanning the retina from left to right and top to bottom. Each black cell of the pattern corresponds to a one in the N-tuple representation, and each white cell to a zero. A similar representation is used for features.

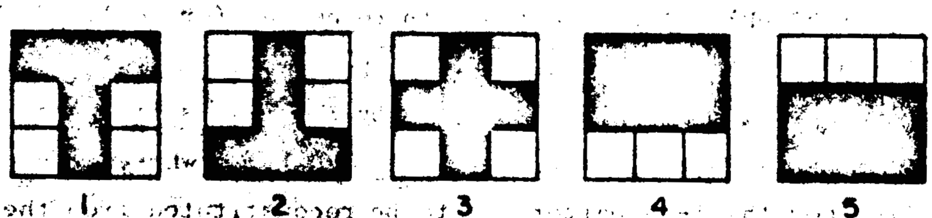
We define the feature matrix $\left[f_{ij}\right]$ as the $K \times N$ matrix of ones and zeros whose i-th row is the N-tuple representing the i-th feature. That is, if $f_{ij} = 1$, the j-th component (or cell) of the i-th feature is one (or black); otherwise the j-th component (or cell) of the i-th feature is zero (or white). The specifications for combining features to form M patterns are given by the composition matrix, $\left[e_{ij}\right]$; $\left[e_{ij}\right]$ is an $M \times K$ matrix of ones and zeros whose i-th row specifies how the i-th pattern is to be reconstituted from the K features. If $e_{ij} = 1$, the j-th feature is used in the reconstitution of the i-th pattern; otherwise the j-th feature is not so used. Reconstitution is accomplished by the superposition of all features used.

The following example illustrates these ideas. As shown on the next page, a set of four features on a 3×3 retina are combined in various ways to form a set of five patterns ($K = 4$, $N = 9$, $M = 5$). The composition matrix and the feature matrix for this example are as follows:

$$\left[\epsilon_{ij}\right] = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad \left[f_{ij}\right] = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$



FEATURES



PATTERNS

This example shows how a set of features can be combined to form a set of patterns. Our problem is the converse: given a set of patterns, find a set of K features from which they can be reconstituted. Given a training set Φ and the number K, it may be impossible to find matrices $\left[\epsilon_{ij}\right]$ and $\left[f_{ij}\right]$ such that each pattern in Φ is reconstituted exactly. Of course, if $K = M$ or N, exact reconstitution can always be made by selecting the features to be either the M patterns themselves or the N distinct patterns with one black cell each. We shall usually be concerned with the case $M > N > K$. If exact reconstitution is impossible with only K features, we desire to find composition and feature matrices yielding reconstituted patterns that are good approximations to the patterns in Φ.

Since we do not expect to be able to find composition and feature matrices that reconstitute the patterns in Φ exactly, we must specify what is meant by approximate reconstitution. Let Φ' be the set of reconstituted patterns produced by a composition matrix and a feature matrix. Let E be the total number of cells in which the patterns in Φ differ from those in Φ'. We shall attempt to find, for any Φ and specified K, composition and feature matrices that minimize E.*

The following algorithm (called the exhaustive iterative algorithm) was developed through consultations with Prof. H. D. Block of Cornell University. For a given training set, Φ, and number K, we iteratively adjust the entries in the epsilon and feature matrices until the total error, E, can no longer be reduced. We begin with a random assignment of ones and zeros for the entries in these matrices.

The adjustments are made as follows: Select an entry, say ϵ_{ij}, of the composition matrix. Compute the effect on E obtained by changing only the value of this entry. If E increases, do not change ϵ_{ij}. If E decreases, make the change. If E remains the same, we have three possibilities:

(a) Always make the change

(b) Never make the change

(c) Make the change with some probability, p.

Each of these possibilities defines a different mode of application of the algorithm. We shall label these as Modes (a), (b), and (c), according to the possibility selected.

---

*This is obviously a very elementary definition of reconstitution error. In lieu of other error measures, it was used throughout this study as a means of comparing the results achieved with the different algorithms. The algorithm to be described can be used with any other error measure that is sensitive to the change of one retinal cell in one pattern.

The algorithm proceeds by applying the above rule to each entry of the composition matrix in turn. After the composition matrix has been thus adjusted, we apply a similar process to the feature matrix. Select an entry, f_{ij}, of the feature matrix and compute the effect on E obtained by changing only the value of this entry. If E increases, do not change f_{ij}. If E decreases, make the change. Otherwise apply Modes (a), (b), or (c).

One pass through all entries of the composition and feature matrices is called an iteration. A successful pair of matrices may result only after several iterations. Note that as the algorithm proceeds, E is monotone non-increasing. Therefore, application of the algorithm results in one of the following three terminal conditions:

(1) It achieves a set of features producing a minimum value of E.

(2) It wanders endlessly on a plateau above the minimum E state.

(3) Or, it is trapped by a local but not absolute minimum of E.

We have observed experimentally that using Modes (a) or (b) and cycling through the composition and feature matrices in fixed order often results in Terminal Condition (2). This terminal condition can be avoided by using Mode (c), but even with Mode (c) the possibility of Terminal Condition (3) has not been ruled out. Even though this algorithm is not guaranteed to achieve a set of features producing a minimum value of E, the use of Mode (a) has generally led to results that were consistent and difficult to improve further.

The results of experimental tests of this algorithm were reported previously[5] and are only briefly summarized here. A set of eighty patterns, each in the form of a solid triangle or the degenerate remnant of such a figure on a 5-by-5 retina, were split into two sets, a set of forty training patterns and a set of forty testing patterns. Composition and feature matrices were determined for the training patterns by using the exhaustive iterative algorithm [Mode (a)] for twenty iterations. The features thus determined were then used to reconstitute the testing patterns; the composition matrix for these patterns was determined by the same iterative adjustment method.

The results obtained depended upon the value chosen for the parameter K, the number of features. For K = 6, 9, and 12, the following average number of errors per pattern were obtained:

K	Average Training Set Errors	Average Testing Set Errors
6	2.25	3.05
9	1.65	2.60
12	1.08	2.03

These errors were reasonably evenly distributed over the patterns, so that the numbers given can be considered to be representative of the number of errors encountered in reconstituting any one pattern. On this basis, the results obtained seemed to be promising.

However, with patterns on such a small retina it is difficult to draw firm conclusions about the quality of reconstitution. The amount of computation time required by the exhaustive iterative algorithm is roughly proportional to NMK^2, and is prohibitively long for experiments using appreciably larger retinas. Thus a conclusive evaluation of the usefulness of feature detection for bandwidth compression can not be done without a faster algorithm. The following section of this report describes other algorithms that were investigated in the hope of finding one that would give comparable performance without being effectively limited to small retinas.

III EXPERIMENTS WITH OTHER ALGORITHMS

The three different algorithms that were investigated were tested on three different sets of patterns. The digital computer programs developed for these experiments were carefully tailored to provide maximum efficiency. The algorithms themselves are described first, followed by a description of the patterns, and a comparison and evaluation of the results of the experiments. A discussion of special features of the computer programs is given in Appendix A. A detailed listing and explanation of the computer runs is given in Appendix B.

A. THE ALGORITHMS

1. Intersections and Unions of Patterns

Elements common to a pair of patterns constitute the intersection of the pair and may be selected by performing a logical AND operation between corresponding elements of the two patterns. Elements included in one or both of a pair of patterns constitute the union of the pair and may be selected by performing a logical OR operation between corresponding elements of the two patterns.

In the B5000 computer, up to 47 binary elements can be accommodated in a single computer word, and the logical operators AND and OR perform their operations on all bits of a pair of words in parallel. Thus the intersection or union of two patterns of 47 bits or less, or of 47-bit areas of two larger patterns, can be accomplished in a single logical operation.

Experiments on the B5000 were conducted using 25- and 35-bit patterns, each pattern occupying a single computer word. Although details varied from experiment to experiment, the essential operations were generally as follows: Certain patterns of the training set were designated as tentative features. The remaining patterns were tested as to the absolute and relative sizes of their intersections with the tentative features. Generally, each of the remaining patterns was combined with that tentative feature with which it made the largest intersection. If that intersection was larger than a critical size (an arbitrary parameter), the tentative feature was replaced by the intersection, which was then the tentative feature for the next round of tests. If the maximum intersection made by a given pattern was smaller than the critical size, the tentative feature was replaced by its union with the pattern. Each pattern of the training set was utilized at least once in arriving at the final set of features.

Various schemes were tried for selecting the initial tentative features, and a wide range of size parameters was investigated. In one form, the algorithm determined the number of features as well as their structures.

2. Simplified Iterative Algorithm

In the exhaustive iterative algorithm described in Sec. II, each element of the feature matrix and each element of the composition matrix was examined in turn and assigned that value which minimized the total error in reconstruction of the pattern set. This process was repeated until the total error stabilized at a minimum value. This procedure was slow, mainly because of the many times the patterns were reconstructed and errors computed.

Several experiments were run on a simpler iterative technique that does not require repeated error evaluation. Arbitrarily formulated initial features were assigned in the composition matrix to the reconstruction of those patterns of the training set which included more than a certain fraction (an arbitrary parameter) of the feature elements. Then the features were modified, element by element, introducing or retaining only those elements that occurred in more than a given fraction (an arbitrary parameter) of the patterns to which that feature was assigned in the composition matrix. This process was repeated several times, first modifying the composition matrix and then modifying the features.

This algorithm was tested with several sets of initial features and a number of choices of the arbitrary parameter.

3. The Sequential Algorithm

This technique for feature determination is described in detail in Ref. 1 (copies of this paper have been forwarded to JPL in connection with this project). The essential operations are as follows:

Each tentative feature is initiated as a filled retina. Sizes of intersections with successive patterns (arbitrary order) are measured. If the size equals or exceeds a certain measure, the intersection becomes the tentative feature. When all of the patterns in the input set have been considered, one feature has been determined in its final form. Repetition of the same process determines additional features.

This process would generate a number of identical features except that the measure used for comparison is equal to the sum of a fixed parameter (arbitrary) and the sum of the retinal elements in the union of all previously determined features that would be totally contained in the modified tentative feature. Thus the measure changes, and successive passes through the pattern set produce distinct features.

After a number of features have been determined, a point will be reached where a complete pass through the pattern set will permit no modification of the full-retina tentative feature. Thus this algorithm determines the number as well as the form of the features.

This algorithm was tested with several types of pattern sets and a number of choices of the arbitrary parameter.

B. THE PATTERNS

Any approach to bandwidth compression must exploit, explicitly or implicitly, some statistical property or properties of the message set. If all possible patterns on a given-sized retina were equally probable and full resolution were required in their transmission, no bandwidth compression would be possible. The basic assumption of the feature detection and utilization approach to bandwidth compression is that patterns of significance to the user at the receiving end of the transmission system form a limited subset with certain common characteristics.

Three distinctly different pattern sets were used for testing the several algorithms for feature detection. There are, of course, a great many statistical measures which could be evaluated for the pattern sets. The question of statistics was not pursued in depth, however, and only the simplest of statistical computations was performed. This consisted of counting the frequencies of occurrence of the retinal elements for each set and the frequencies of element pairs (digram frequencies). Although no quantitative conclusions have been drawn from these statistics, they provide some insight useful in the interpretation of some of the feature-detection results.

1. Triangular Pattern Set

This is the set of 80 patterns utilized in the experiments described in Ref. 5 (and reproduced in Figs. 1 and 2 of that report). They were constructed according to a rigid rule and exhibit a high degree of regularity. Each pattern is a solid triangle or the degenerate remnant of such a figure.

The patterns were constructed by considering each element of the 5-by-5 retina as the apex of four right triangles, with bases up, down, right, and left. Each of the four solid triangles was considered as a pattern, even if the apex lay at an edge of the retina and only a single element was retained. One hundred such patterns may be identified, twenty of which are redundant, leaving a total of 80 distinct patterns. The number of elements per pattern varies over a wide range. Sixteen patterns consist of single elements on the retinal perimeter; four contain three elements each; twelve contain four elements each; etc. The largest patterns are eight of 18 elements and four of 19 elements.

Variation among element frequencies in the triangular pattern set are small. The array of element frequencies (Table I) is completely symmetric, reflecting the symmetry of the rule by which the patterns were constructed.

Table I

ELEMENT FREQUENCIES--TRIANGULAR PATTERNS (80)

25	26	26	26	25
26	29	30	29	26
26	30	32	30	26
26	29	30	29	26
25	26	26	26	25

Digram statistics of this pattern set are similarly bland. In the following arrays (Table II), the boxed number designates the element to which the digram frequencies refer and is the total frequency of that element. The other entries are the numbers of times those elements occur together with the boxed element in the total pattern set.

Table II

DIGRAM FREQUENCIES--TRIANGULAR PATTERNS (80)

$\boxed{25}$	20	16	12	9	20	$\boxed{26}$	20	16	12	16	20	$\boxed{26}$	20	16
20	23	17	12	7	18	21	22	15	10	13	20	21	20	13
16	17	17	10	5	13	16	16	14	7	11	14	16	14	11
12	12	10	8	3	10	10	10	8	6	7	9	9	9	7
9	7	5	3	2	7	6	4	4	3	5	4	4	4	5

23	21	20	15	12	17	22	21	22	17	17	16	16	16	17
21	$\boxed{29}$	22	16	10	16	22	$\boxed{30}$	22	16	16	23	23	23	16
20	22	23	15	9	14	22	23	22	14	16	23	$\boxed{32}$	23	16
15	16	15	14	8	14	15	16	15	14	16	23	23	23	16
12	10	9	8	8	10	10	9	10	10	17	16	16	16	17

The digram frequencies are seen to decrease rather monotonously with distance between pairs of elements. There are no surprises in the statistics.

The triangular patterns are low-detail figures which could be tranmitted without serious loss of information by low-resolution analog techniques. They appear to be essentially featureless, in the sense that their statistics are smooth and uninteresting.

An approximation to simple resolution reduction was tested with this pattern set. Nine non-overlapping features, consisting of clusters of three retinal elements each except for a lone central element, were assigned to reconstruction of the patterns. The features were:

```
xx...  ...xx  .....  .....  .....  ..x..  .....  .....  .....
x....  ....x  .....  .....  .x...  ..xx.  .....  .....  .....
.....  .....  .....  .....  xx...  .....  ...xx  .....  ..x..
.....  .....  ....x  x....  .....  .....  ...x.  .xx..  .....
.....  .....  ...xx  xx...  .....  .....  .....  ..x..  .....
```

These triangular clusters are as close an approximation as can be made on this small retina to a simple reduction of resolution by a factor of about three. The features were not modified in this experiment but simply assigned to the reconstruction of the patterns so as to minimize the total errors for the training and testing sets. The errors obtained were:

Training Set Errors		Testing Set Errors	
Total	Average	Total	Average
76	1.90	88	2.20

Since there was no training, the difference between the results for the two sets is purely due to the arbitrary division of the 80 patterns into two groups. The average error per pattern is 2.05 elements, resulting from 8 patterns with zero error, 16 with one error, 20 with two errors, and 36 with three errors.

2. Alphanumeric Pattern Set

To be useful, a feature-detection algorithm should provide reasonable performance on different types of pattern sets. To test the various tentative algorithms on patterns composed of line segments and to avoid the regularity of patterns contrived by the operation of rigid rules, we selected a set of alphanumeric characters designed on a 5-by-7 matrix for use with electrical and mechanical printers. The set consisted of 35 characters (A through Z, all capitals, and 1 through 9) as produced by the print wheels on the IBM keypunch.

Element frequencies (Table III) showed wide variation in the alphanumeric pattern set, and there appears to be little symmetry to the distribution.

Table III

ELEMENT FREQUENCIES--ALPHANUMERIC PATTERNS (35)

20	19	23	21	17
24	7	4	3	23
21	7	7	6	18
18	13	20	13	13
20	8	10	5	18
25	4	5	6	18
16	20	23	18	14

The main feature apparent in the element-frequency array is the concentration of large numbers around the perimeter and horizontally across the center of the retina.

Digram statistics of the alphanumeric patterns show some long-range correlations and interestingly low short-range correlations in some directions. A few of the diagram arrays (Table IV) will illustrate this.

RE-ORDER No. 64-123

Table IV

DIGRAM FREQUENCIES--ALPHANUMERIC PATTERNS (35)

$\boxed{20}$	10	10	10	15	10	$\boxed{19}$	19	19	7	10	19	$\boxed{23}$	20	7
15	4	1	2	14	14	2	1	0	14	14	5	3	1	14
13	5	5	5	10	11	4	2	3	10	13	4	4	3	11
11	9	12	6	7	8	9	13	8	6	10	10	16	9	7
11	6	5	2	8	10	4	3	1	9	12	5	6	2	11
14	4	3	3	9	14	3	1	3	9	16	3	3	3	11
14	8	10	7	10	8	15	15	12	6	9	17	18	14	7

10	19	20	$\boxed{21}$	7	15	7	7	7	$\boxed{17}$	15	14	14	14	13
14	3	2	1	14	13	2	1	2	11	$\boxed{24}$	2	0	2	18
12	5	2	4	10	11	3	5	5	7	19	4	4	3	14
10	10	14	10	6	9	5	9	4	8	15	10	13	10	10
12	5	4	2	11	9	4	4	2	8	17	2	5	3	12
15	3	1	4	10	13	2	3	2	9	21	1	2	5	13
8	16	16	14	6	9	7	9	6	9	12	13	15	11	12

4	2	5	3	2	1	1	3	2	1	2	0	1	0	2
2	$\boxed{7}$	1	2	4	0	1	$\boxed{4}$	0	0	2	2	0	$\boxed{3}$	1
4	2	3	0	5	0	1	3	1	0	3	0	2	0	2
4	3	4	3	4	1	0	3	1	0	3	1	1	0	2
4	3	2	1	6	1	1	4	1	1	3	1	2	1	2
4	2	1	0	6	0	0	3	1	0	3	0	0	1	2
5	4	4	4	3	0	1	3	2	0	3	0	0	0	3

14	14	14	14	11	13	11	13	12	11	5	4	4	5	3
18	4	0	1	$\boxed{23}$	19	4	0	3	13	4	2	1	0	5
13	5	2	4	17	$\boxed{21}$	1	4	1	13	1	$\boxed{7}$	1	4	2
10	9	14	10	10	17	10	9	9	11	1	2	4	2	2
12	6	4	3	13	18	2	5	2	13	1	4	2	3	4
17	4	2	4	14	19	1	1	5	12	3	2	1	1	5
11	14	15	11	9	11	11	12	9	11	3	4	5	5	1

By contrast with the digram statistics of the triangular pattern set, these arrays show wide variations and frequent irregularities. Although the success of feature detection and utilization for bandwidth reduction will depend on higher-order statistics, this statistical behavior at least indicates somewhat greater promise for success than that of the triangular patterns.

Use of the alphanumeric patterns provided some additional insight into the problem of error criteria. In particular, many of the patterns could not be recognized unambiguously although reconstructed with few elements in error, whereas others were recognizeable although perturbed by many errors. Twelve of the 35 characters differ from some other character of the set by only three retinal elements. One, the numeral 4, differs by at least 16 elements from all of the other characters. Thus if character recognition were intended, the simple error criterion employed in these experiments would be totally inadequate.

3. Seven-Feature Pattern Set

A third pattern set used in the evaluation of feature-detection algorithms was previously employed in experiments reported by Block, Nilsson, and Duda.[1] (Figure 14 of Ref. 1 illustrates the complete set.) Constructed on a 5-by-5 retina, the set consists of 22 distinct patterns and two repeats, for a total of 24 patterns. All the patterns can be constructed from some combination of seven features, the simplest feature set being the following:

```
xxxxx  .....  .....  x....  ..x..  ....x  ....x
.....  .....  .....  x....  ..x..  ....x  ...x.
.....  .xxx.  .....  x....  ..x..  ....x  ..x..
.....  .....  .....  x....  ..x..  ....x  .x...
.....  .....  xxxxx  x....  ..x..  ....x  x....
```

The measured statistics of these patterns are shown in Tables V and VI.

Table V

ELEMENT FREQUENCIES--SEVEN-FEATURE PATTERNS (24)

21	15	18	15	22
14	0	7	6	12
14	7	19	7	12
14	6	7	0	12
22	14	18	14	18

Table VI

DIGRAM FREQUENCIES--SEVEN-FEATURE PATTERNS (24)

$\boxed{21}$	15	16	15	20	15	$\boxed{15}$	15	15	15	16	15	$\boxed{18}$	15	17
14	0	5	5	11	8	0	4	3	7	9	0	7	4	8
14	7	17	7	11	8	6	13	6	7	9	6	16	6	8
14	5	5	0	11	8	3	4	0	7	9	4	7	0	8
20	13	15	13	17	14	10	12	10	12	16	11	15	11	13
15	15	15	$\boxed{15}$	15	20	15	17	15	$\boxed{22}$	14	8	9	8	13
8	0	4	3	7	13	0	6	6	12	$\boxed{14}$	0	2	2	9
8	6	13	6	7	13	7	18	7	12	14	6	10	6	9
8	3	4	0	7	13	6	6	0	12	14	2	2	0	9
14	10	12	10	12	21	13	16	13	17	14	8	9	8	12
5	4	7	4	6	5	3	4	3	6	11	7	8	7	12
2	0	$\boxed{7}$	1	3	2	0	1	$\boxed{6}$	1	9	0	3	1	$\boxed{12}$
2	0	7	0	3	2	0	6	0	1	9	5	9	5	12
2	1	7	0	3	2	6	1	0	1	9	1	3	0	12
5	3	7	3	4	6	2	3	2	3	12	8	9	8	12

14	8	9	8	13	7	6	6	6	7	17	13	16	13	18
14	0	2	2	9	6	0	0	0	5	10	0	7	6	9
$\boxed{14}$	6	10	6	9	6	$\boxed{7}$	7	7	5	10	7	$\boxed{19}$	7	9
14	2	2	0	9	6	0	0	0	5	10	6	7	0	9
14	8	9	8	12	7	4	4	4	6	17	9	13	9	13

Et Cetera--

Some of the features appear prominently in the digram arrays. When the reference element in a digram array occurs only in a single feature, all elements of that feature exhibit the maximum value for the array. Other elements which also exhibit this maximum value may be included in that feature, since they always occur with the other elements in the pattern set. This situation occurs in the ninth digram array, where the right-hand column and the lower left-hand corner display the value 12, and also in the eleventh array, where the three-element central horizontal bar and three corner elements display the value 7. Thus from these simple statistics we can see that the feature set is not unique.

C. COMPARISON AND EVALUATION OF PROMISING ALGORITHMS

Each of the techniques for feature detection was tested in several modified forms and usually with a range of values for each of the arbitrary parameters involved. About 70 runs were made on the B5000 computer, and many of these tested more than one technique or several values of some parameter.* Some procedures were common to all or most of the runs, and the general order of business was as follows:

---

*A detailed listing and explanation of the computer runs is given in Appendix B.

A pattern set was read into memory. In some cases, this pattern set was treated as two subsets, a training set and a testing set. Each pattern may be considered as a 5-by-5 or 5-by-7 binary array, although stored as a single word in the computer. A block of memory was set aside for the features to be determined. Each feature is essentially a pattern on the same retinal array as the input patterns. In some cases, a set of initial features was loaded in the input process; in other cases, the feature set was initially empty. Another block of memory was reserved for the composition matrix, which may be considered as a rectangular array with one dimension equal to the number of features and the other dimension equal to the total number of patterns (training and testing). A binary 1 occupying the element corresponding to a given feature and a given pattern assigns that feature to the reconstruction of that pattern. The reconstructed pattern is then the union of all features so designated in the composition matrix. The composition matrix could be loaded with tentative values at input, or considered initially empty.

Various printouts were provided during performance of some of the feature-detection algorithms, so that details of the process could be reconstructed and analyzed. In other cases, only the final result was printed. In iterative techniques that determined features and composition matrix together, pattern errors were printed out at each iteration so that the trend toward or away from convergence could be examined.

Following the performance of algorithms that produced features alone, the composition matrix was determined by performing several iterations of the procedure used for modification of the composition matrix in the exhaustive iterative algorithm as described in Sec. II.

In each experiment, pattern errors were evaluated by comparing the reconstructed patterns with their input counterparts, and totalling errors for the set. If there was a subset of testing patterns, the above procedures for determining the composition matrix and errors were always employed for this set.

Since the exhaustive iterative algorithm described in Sec. II served as a standard of comparison, one evaluation technique which was frequently employed consisted of running this algorithm in tandem with the scheme being evaluated. The features produced by the algorithm under test were used as initial features for the iterative procedures, and the disparity between the error figures obtained with the initial and final sets of features was considered as the margin for possible improvement.

In most cases, printout included the input patterns, features, and composition matrix (unless empty), features produced by the algorithm under test, final composition matrix, and reconstructed patterns with a tabulation of errors, together with whatever special information was produced during the feature-detection process.

As a standard for comparison, the exhaustive iterative algorithm appears to give fairly consistent results, although absolute minimum errors are not guaranteed by this technique. The following sets of results were obtained on the Triangular Patterns when the exhaustive iterative algorithm was asked to find nine features:

	Training Set Errors		Testing Set Errors	
	Total	Average	Total	Average
(a)	66	1.65	104	2.60
(b)	66	1.65	86	2.15
(c)	65	1.63	88	2.20

Results (a) were obtained on the IBM 7090 and in Run 21 on the B5000 using the same set of arbitrarily constructed initial features and the same arbitrary initial composition matrix.* Results (b) were obtained in Run 24 using a set of three-element clusters as initial features,

---

*The computer runs are listed in Appendix B.

with the composition matrix initially empty. Results (c) were obtained in Runs 35 and SRI 4 and 5, which employed one of the "intersections and unions" algorithms to produce a set of initial features.

Performance on the training set is seen to be quite consistent, although it is evident that the features determined depend somewhat on the initial set. The primary objective of the work herein reported was to develop an algorithm which would approach this error performance with much less computation.

Operating on the Alphanumeric Pattern Set, the exhaustive iterative algorithm produced the following results:

	Features	Pattern Set Errors Total	Pattern Set Errors Average
(a)	8	136	3.89
(b)	8	152	4.34
(c)	8	155	4.43
(d)	9	144	4.11

These results were obtained from initial features produced by different versions of the "intersections and unions" algorithm, Runs 39, 40, and 41, and SRI 4 and 5. Nine features were found in (d), as compared with eight in the others, so that one would expect a lower minimum error for this case. Scatter of the results implies that further experimentation might well yield lower minima.

Operating on the Seven-Feature Pattern Set, the exhaustive iterative algorithm failed to yield seven features that would reduce the error to zero. Initial features, derived from one of the "intersections and unions" techniques, yielded errors of 36 bits total, or 1.50 bits per pattern on the average. The exhaustive iterative algorithm reduced this error to 17 bits total, or 0.71 bit average, but could do no better. The initial features were distinct and reasonable, providing no easy explanation for the failure to go all the way to the known minimum possible error, namely zero.

1. Intersections and Unions of Patterns

Techniques utilizing intersections and unions of patterns to determine features evolved through many modifications. Initial effort was devoted to tests utilizing the Triangular patterns. A totally random scheme for selecting patterns as tentative features and then modifying them by intersections (if larger than an arbitrary size) or unions with randomly selected members of the training pattern set yielded large errors. The scheme was then modified to select the first few patterns as tentative features and combine them with those patterns with which they had the most elements in common. The best results with the Triangular patterns were obtained using the largest patterns as tentative features and combining them with the remaining patterns with which they made the largest intersections.

Multiple runs were made to determine the optimum value for the arbitrary parameter determining whether an intersection or union would be taken between a tentative feature and the pattern with which it shared the most elements. Six, nine, and twelve features were determined in these runs, in the hope of establishing an empirical relationship for determining the size parameter. In the following tabulation, values in parentheses are minimum errors produced by the exhaustive iterative algorithm, included here for comparison

Features	Size Parameter	Training Set Errors Total	Training Set Errors Average	Testing Set Errors Total	Testing Set Errors Average
6	6	111	2.77	130	3.25
		(90)	(2.25)	(122)	(3.05)
9	4	76	1.90	113	2.83
		(65)	(1.63)	(88)	(2.20)
12	3	74	1.85	90	2.25
		(43)	(1.08)	(81)	(2.03)

The product of the number of features and the optimum size parameter, as determined by these data, is 36 in each case, leading to a tentative relationship: $F \cdot S = (x + 1) \cdot (y + 1)$, where F is the number of features, S is the size parameter, x and y are the linear dimensions of the retina.

After selection of the tentative features, the rest of the patterns were considered in the order in which they were read into memory. Six runs were made to check sensitivity to order of input, with the data cards (one per pattern) shuffled between runs. Mean errors and root-mean-square deviations for these six runs were:

	Training Set Errors Total	Training Set Errors Average	Testing Set Errors Total	Testing Set Errors Average
Means	82.83	2.07	113.33	2.83
RMS Deviation	3.98	0.10	8.44	0.20

Thus for this pattern set, the order of the patterns is not very important.

This algorithm was not very successful with the Alphanumeric or Seven-Feature pattern sets. In the following, the numbers in parentheses are the errors resulting after employing the exhaustive iterative algorithm in tandem with the feature-detection algorithm under examination:

Pattern Set	Features	Pattern Set Errors Total	Pattern Set Errors Average
Alphanumeric	9	183 (144)	5.23 (4.11)
Seven-Feature	7	36 (17)	1.50 (0.71)

Numerous modifications to this algorithm were made in the course of experiments with the Alphanumeric Pattern Set. It appeared that the large patterns selected as tentative features might be too similar to provide the best starting set, so various techniques were employed to prevent patterns with too much overlap from being selected as initial tentative features. The most successful of these eliminated a pattern from consideration as a tentative feature if more than three-quarters of its elements were included in any previously-chosen tentative feature. The number of tentative features which had been chosen when the pattern set was exhausted became the number of final features determined.

Although exhibiting improved performance with the Alphanumeric patterns, this version of the algorithm failed completely with the other pattern sets, as the following data show:

Pattern Set	Features	Pattern Set Errors Total	Pattern Set Errors Average	
Alphanumeric	8	167	4.77	
		(136)	(3.89)	
Triangular	2	225	5.63	(Training)
		226	5.65	(Testing)
Seven-Feature	2	172	7.17	

Failure to perform with the Triangular and Seven-Feature pattern sets is simply due to the fact that a combination of two patterns, in each case, included all or nearly all of the elements contained in all of the other patterns of the set. Clearly, this approach to improvement of the algorithm, while reducing errors for one pattern set, was of no general value.

As instrumented on the B5000, unions and intersections of patterns can be performed quickly, and hence, cheaply. Error evaluation was not involved in the process of determining features, so that this somewhat time-consuming operation was required only for final evaluation. Run time for these algorithms was much less than with any of the iterative techniques.

2. Simplified Iterative Algorithm

This technique, described in general terms in Sec. III-A-2, was investigated using the Triangular Pattern Set and two sets of initial features. Four computer runs (Runs 47, 48, 49, 50) explored a wide range of parameter values. Runs utilizing the cluster set of initial features consistently diverged; i.e., errors increased with successive iterations. In all but one case, duplicate features were produced. The number of distinct features varied from three to six, although nine initial features were employed. In none of these experiments could the technique be said to converge.

Failure of this technique may be ascribed to the fact that each feature is considered independently of all others and evolved from identical operations utilizing the same pattern set. There is no mechanism to guarantee a diversity of features which would cooperate for optimum reconstruction of the patterns.

3. <u>The Sequential Algorithm</u>

This technique, discussed in Sec. III-A-3, was investigated with all three pattern sets. The parameter determining minimum feature size was varied from unity to five or six. This algorithm determines the number of features, so that only an upper limit on this number was entered into the system initially.

Performance on the Seven-Feature Pattern Set was as follows:

Size Parameter	Features	Pattern Set Errors Total	Pattern Set Errors Average
1	14	0	0.00
2	14	15	0.63
3	14	18	0.75
4	10	31	1.29
5	6	43	1.79

In the first case, seven features adequate for complete reconstruction of the patterns were found, along with an equal number of redundant fragments. It was disappointing that, as the size parameter was increased, the features determined would no longer provide error-free reconstruction, since it is known for this set that seven features, none smaller than five elements, will suffice for perfect reconstruction.

With the Triangular patterns, a value of unity for the size parameter produced 25 single-element "features"--the 25 retinal elements. With the size parameter as large as six, eleven features were being determined, and the error rate exceeded that of other techniques for nine features.

An upper limit of 20 features was used in experiments with the Alphanumeric Pattern Set, and with the minimum feature-size parameter as large as five, 20 features were being determined.

Features found by this algorithm tended to be equal in size or only slightly larger than the minimum size parameter. This tendency apparently resulted in the determination of a number of feature fragments which were redundant in the reconstruction process.

IV CONCLUSIONS

The basic premise upon which this investigation was based was the following: the pictures of interest to the recipients can be adequately approximated by the superposition of certain patterns called <u>features</u>, the number of features being significantly smaller than the number of retinal cells. If this premise is valid, its exploitation for bandwidth compression requires a practical procedure for determining suitable features from a representative set of patterns, and for detecting their presence or absence in any given pattern. Accordingly, this study was devoted to finding and evaluating such procedures.

The results of this research were largely negative. Although the experiments performed were limited to patterns on very small retinas, making it difficult to draw firm conclusions about the quality of performance, the following observations can be made:

(1) The only algorithm found that achieved uniformly good results on a variety of pattern sets was the exhaustive iterative algorithm. However, even this algorithm did not always determine a minimum-error feature set; for example, different limiting errors were obtained when different initial features were used.

(2) The amount of computation required by the exhaustive iterative algorithm limits its use to small retinas. Simpler algorithms, investigated in the hope of minimizing computation time, resulted in higher error rates. None performed sufficiently well on a variety of pattern sets to warrant experiments using large retinas.

(3) Of the simpler algorithms studied, two techniques based on taking intersections and/or unions of the

training patterns might merit further investigation for specific applications. Each would require modification and specialization to the application at hand.

These results neither prove nor disprove the premise that feature detection is a potentially useful bandwidth-compression technique. However, none of the algorithms proposed for feature determination was found satisfactory. Moreover, the results give little indication of how better algorithms might be developed. In the absence of such information, it is recommended that the study of feature detection for bandwidth compression be terminated.

In conclusion, the authors would like to make some observations which are not directly related to the results of this investigation, but which nevertheless seem pertinent to this study. Television or facsimile transmission of monochrome pictures generally treats each resolvable element of each and every picture as an independent brightness element. Any hope for compression of the bandwidth-time-power product in the transmission of a series of such pictures must be based on some hypothesized or established property (or properties) of the input pictures, or on some restriction on the information required at the receiving station.

In television transmission, if successive frames are highly correlated and if precise reproduction of individual frames is not required, frame-to-frame redundancy may be exploited for bandwidth compression. This is essentially a time-bandwidth trade-off. Such techniques might be useful for real-time monitoring of soft landings or remote guidance of slowly moving vehicles. Instrumentation may be expected to be quite complex.

Where individual frames are not highly correlated, as in the case of programmed snap-shot photography from a fast-moving vehicle, each picture must be considered independently. The possibility of bandwidth compression in such cases depends not so much on the nature of an individual picture as on the statistics of the ensemble of possible pictures and the degree of detail important to the recipient.

At one extreme lies the case where all possible patterns obtained with the full resolution capability of the camera are equally probable, and all detail is of interest in analysis of the picture. Exploration of unknown terrain, where a priori information is minimal, would appear to be an example of this sort. In such cases there would appear to be little hope for bandwidth compression unless the recipient is willing to settle for something less than full resolution, full gray scale, or full picture area.

At the other extreme lies the case where the input pattern set is limited to a relatively small number of patterns, or the recipient's requirement for information is limited to pattern classifications. For example, the transmission of alphanumeric patterns only requires transmitting an identifying code word if pattern-recognition capability is available at the transmitter. Whenever classification into a relatively few categories supplies the required information, pattern recognition offers an effective means of reducing bandwidth. Feature detection may play a much more valuable role in such applications.[1]

Between the two extremes undoubtedly lie many cases of interest in which either the input picture set is somehow restricted or the recipient's information requirement is such that precise reproduction of the input picture is not required. Most probably, efficient transmission will be achieved in most of these cases by employing specialized techniques that depend on the specific restrictions on the input pattern set or on the detailed nature of the information that is to be abstracted from the transmitted signal. Techniques may range all the way from simple reduction of resolution for low-detail pictures to sophisticated schemes for pattern recognition. It is thus difficult to generalize about the many bandwidth-compression possibilities applicable to special circumstances. Their characteristics and their effectiveness will depend on the specifics of each information-transfer problem.

The general problem is not being neglected, however. Considerable motivation for pursuing more-or-less general approaches to bandwidth compression stems from the intuitive notion that pictures that "make sense" must form a restricted set, that the great percentage of "all possible" pictures must be senseless noise patterns and, as such, should not occur with great frequency when one looks at structures created by the organized processes of man or nature. This is an entirely reasonable point of view. This premise leads to a search for properties that differ significantly between pictures that "make sense" and those that do not, properties that can be measured, manipulated, and exploited for transmission economy by available technology. This search has been under way for a number of years and is still an active area of investigation. To date, limited success has been achieved, but results are well short of the hopes of the investigators.

"Run-Length Coding" may be cited as an example of limited success in finding a statistical parameter that can be exploited, fairly generally, for bandwidth compression in television or facsimile transmission. Schreiber and Knapp[6] have demonstrated that for a wide range of pictorial material the transmission of a black-and-white picture (or the most significant digit of a PCM-coded half-tone picture) can be achieved with a bandwidth compression of about four to one by transmitting digitally-encoded run lengths (i.e., distances between black-white or white-black transitions). This work was done in one dimension, along scanning lines of a TV raster. Effort is under way to extend this concept to two dimensions.

Run-length coding exploits a particular statistical property of pictures that "make sense." The ensemble of "all possible" pictures may be considered as the output of a random-noise generator spewing out a stream of independent black and white picture elements. The probability of a run of successive black or successive white elements in a TV line of specified length, as produced by this random generator, is a rapidly-decreasing function of the length of the run. Success of run-length coding is direct evidence that in pictures with sufficient organization to be meaningful, the run-length distribution differs from that of the random-element generator. It is this statistical difference that makes possible a bandwidth saving.

Any solution to the general problem of bandwidth compression for the transmission of pictures, where a reasonably-precise reproduction of the input is required, must exploit some such statistical property, either explicitly or implicitly. Since an endless number of statistical measures can be invented, the hope remains that techniques can be found for significant compression of the bandwidth-time-power product in the transmission of pictures.

Whatever techniques become available in the future, certain basic principles will always be true. These principles are central to the solution of specific problems within the limitations of currently available technology as well as to an attack on the general problem. Whatever the detailed technique applied, bandwidth economy in picture transmission will be achieved most effectively by (1) making the best possible use of all available a priori information, (2) transmitting only that information required in the subsequent analysis of the pictures, and (3) exploiting whatever statistical properties may be common to the class of meaningful pictures.

APPENDIX A

COMPUTER PROGRAMS FOR EXPERIMENTS ON FEATURE DETERMINATION AND PATTERN RECONSTITUTION

Two features available through Extended Algol on the Burroughs B5000 computer make this machine particularly attractive for the manipulation of binary patterns. Partial word addressing permits the packing of up to 47 logical values per computer word and provides the means for modifying such words bit by bit. Logical operations on these words operate on all bits in parallel, facilitating rapid construction of intersections and unions of patterns.

The attractiveness of these features and the availability of free time on the Stanford University machine during its checkout period provided impetus for the programming of a number of experiments on feature determination and pattern reconstitution using the B5000. Efficiency of the machine and its programming language were demonstrated early in these experiments by running a problem which essentially duplicated a previous problem programmed in Balgol and run on the IBM 7090. Although the B5000 is basically slower than the 7090 by a factor of two or more, a problem in which 25-element patterns were manipulated as single words in the Burroughs machine ran in about one-sixth the time required by the 7090.

Extended Algol on the Burroughs B5000 proved to be a very convenient and flexible language for programming these experiments. All of the runs reported were made within the framework of a single basic program. Quantities which varied from experiment to experiment, such as retinal size (25 or 35 elements), numbers of patterns in the training and testing sets, number of features to be sought, and the numbers of iterations to be performed, were entered as data from a single card at the head of the data deck. These data established array sizes and controlled all repetitive operations, utilizing the computer efficiently for either large or small problems. With the dynamic storage allocation of the B5000, several data decks could be stacked to execute separate experiments in a single run, and each experiment would run as efficiently as if separately programmed.

One-dimensional arrays of computer words served as two-dimensional arrays of binary elements for storage of input and reconstructed patterns, features, composition matrix, and an error matrix. A single input procedure served to read these data (one pattern, feature, or matrix row per card) and pack the elements into computer words. Another procedure reversed this process and printed out the patterns, features, or matrices as required.

Operations which were repeated on demand by the various algorithms, such as pattern reconstruction, error evaluation, summing of columns or rows of the various matrices, etc., were written as separate procedures. Since some of these procedures were called as many as 20,000 times in a given experiment (exhaustive iterative algorithm), they were written to achieve as efficient execution as possible. The library of such procedures may be viewed as a sub-language in which the various algorithms were programmed. For instance, the single word "RECON" would call a procedure which reconstructed the Ith pattern, and the one-word statement "ERROR(1,M)" would reconstruct the first M patterns (using "RECON"), compare them with their input patterns, and compute, store, and print out the pattern errors.

Programming of the feature-detection algorithms was simplified and shortened by the use of this sub-language of often-repeated procedures. The final technique programmed, for instance, the "Sequential Algorithm," required the keypunching of only 22 additional cards to be inserted into the master deck. As a consequence, results of the first experiment on this algorithm were available on the day following the decision to program it.

The several algorithms were written as procedures which could be added to or removed from the basic program deck as desired. Thus, setting up an experiment involved selecting the procedures to be used, arranging the appropriate set of procedure call cards, and assembling one or more data decks. It was a simple matter to find a set of features by one of the simpler algorithms, assess their performance in reconstructing the pattern set, and then employ them as initial features with the exhaustive iterative algorithm, all in a single experiment.

Multiple experiments in which several values of a parameter were employed in a given algorithm, with the same input patterns, were accomplished by short loops around those procedure calls required to repeat the algorithm and evaluate its performance. Runs involving different pattern sets employed a loop around the whole program so that arrays of the proper sizes could be redefined.

APPENDIX B

SUMMARY OF COMPUTER RUNS

The computer runs are summarized in Table B-I. Information in the table is amplified in the succeeding pages.

Run numbers were assigned chronologically. Some runs are reported out of their chronological order in the following summary, where the information they provided was closely related to or helped to interpret runs made at an earlier time.

a. <u>IBM 7090 Runs - Exhaustive Iterative Algorithm</u>

Completed in the first half of this project, these runs served as the impetus and take-off point for work in the second half.

The Triangular Pattern Set was utilized in all of these experiments, divided into two subsets of 40 patterns each. An arbitrary set of features and a randomly-produced composition matrix were entered as inputs. Composition matrix and features were iteratively modified, element by element, using the rule that if the error were insensitive to a given element, the value of that element was reversed.

<u>6 Features</u>

Training pattern errors, 90 bits total, 2.25 bits average
Testing pattern errors, 122 bits total, 3.05 bits average

<u>9 Features</u>

Training pattern errors, 66 bits total, 1.65 bits average
Testing pattern errors, 104 bits total, 2.60 bits average

<u>12 Features</u>

Training pattern errors, 43 bits total, 1.08 bits average
Testing pattern errors, 81 bits total, 2.03 bits average.

Table B-I

SUMMARY OF COMPUTER RUNS

Run Number	Type of Algorithm	Pattern Set	Number of Features	Train Set Errors Total	Train Set Errors Average	Test Set Errors Total	Test Set Errors Average	Comments
IBM 7090	Iterative, Composition and Features	Triangular	6	90	2.25	122	3.05	
IBM 7090	Iterative, Composition and Features	Triangular	9	66	1.65	104	2.60	Approximate run time on 7090, 30 minutes
IBM 7090	Iterative, Composition and Features	Triangular	12	43	1.08	81	2.03	
17	Iterative, Composition and Features	Triangular	9	126	3.15	175	4.38	Status quo if error insensitive to an element of a feature or of the composition matrix.
23	Iterative, Composition and Features	Triangular	9	77	1.93	116	2.90	Element included in alternate iterations if error insensitive.
21	Iterative, Composition and Features	Triangular	9	66	1.65	104	2.60	Element changed if error insensitive. This rule used in all later runs.
24	Iterative, Composition and Features	Triangular	9	66	1.65	86	2.15	Cluster features initially.
59	Composition Matrix Assignment only	Triangular	9	76	1.90	88	2.20	Non-overlapping Cluster Features. No iteration. Equivalent to simple reduction of resolution.
25	Intersections and Unions	Triangular	9	158	3.95	191	4.77	Random combinations of patterns by unions and intersections.
27	Intersections and Unions	Triangular	9	98	2.45	128	3.20	First nine patterns combined with rest of patterns on basis of maximum intersection.
28	Intersections and Unions	Triangular	9	87	2.18	119	2.98	Largest nine patterns combined with rest of patterns on basis of maximum intersection.
29, 30	Intersections and Unions	Triangular	9	109	2.73	113	2.83	Size Parameter: 3 Parametric study of size parameter
			9	76	1.90	113	2.83	4 Algorithm same as in Run 28.
			9	79	1.98	106	2.65	5
			9	87	2.18	119	2.98	6
			9	87	2.18	119	2.98	7
			9	99	2.48	128	3.20	8
			9	103	2.58	139	3.48	9
			9	112	2.80	154	3.85	12
			9	140	3.50	176	4.40	15
31, 33	Intersections and Unions	Triangular	6	132	3.30	168	4.20	Size Parameter: 2 Parametric study of size parameter.
			6	144	3.60	179	4.48	3 Algorithm same as in Run 28.
			6	118	2.95	131	3.27	4
			6	126	3.15	126	3.15	5
			6	111	2.77	130	3.25	6
			6	123	3.08	138	3.45	7
			6	126	3.15	157	3.93	8
			6	119	2.98	139	3.48	9
			6	119	2.98	140	3.50	10
			6	122	3.05	152	3.80	11
			6	122	3.05	152	3.80	12

4

Run Number	Type of Algorithm	Pattern Set	Number of Features	Train Set Errors Total	Train Set Errors Average	Test Set Errors Total	Test Set Errors Average	Comments
32, 34	Intersections and Unions	Triangular	12	81	2.02	105	2.63	Size Parameter: 2 Parametric study of size parameter.
			12	74	1.85	90	2.25	3 Algorithm same as in Run 28.
			12	78	1.95	101	2.52	4
			12	78	1.95	101	2.52	5
			12	76	1.90	117	2.93	6
			12	76	1.90	117	2.93	7
			12	106	2.65	137	3.43	8
			12	98	2.45	136	3.40	9
			12	96	2.40	144	3.60	10
			12	96	2.40	144	3.60	11
			12	96	2.40	144	3.60	12
35	Intersections and Unions plus Iterative	Triangular	9	65	1.63	88	2.20	Same as above runs, with size parameter 4, followed by Iterative algorithm, as in Run 21.
36	Intersections and Unions plus Revision	Triangular	9	76	1.90	113	2.83	Unions and Intersections as above.
				83	2.08	102	2.55	First revision, unions and intersections with pattern set
				85	2.13	101	2.52	Second revision
				85	2.13	101	2.52	Third revision
				85	2.13	101	2.52	Fourth revision
				85	2.13	101	2.52	Fifth revision
				85	2.13	101	2.52	Sixth revision
37	Intersections and Unions Iterative	Alphanumeric	8	75	3.75	122	7.47	20 Training, 15 Testing. Composition matrix modification.
				58	2.90	117	7.80	Features modified by training set.
39	Intersections and Unions Iterative	Alphanumeric	8	186	5.31			All 35 Training. Eliminates some patterns as tentative features.
				136	3.89			Features modified iteratively.
40	Intersections and Unions Iterative	Alphanumeric	8	217	6.20			Different basis for eliminating some patterns.
				152	4.34			Features modified iteratively.
41	Intersections and Unions Iterative	Alphanumeric	8	167	[illegible].77			Pattern eliminated if intersects more than 3/4 of size.
				155	4.[illegible]3			Features modified iteratively.
43	Intersections and Unions	Alphanumeric	8	167	4.77			Number of features determined by running out of patterns.
44	Intersections and Unions	Alphanumeric	10	194	5.54			Considers all patterns in order of decreasing size.
45	Intersections and Unions	Alphanumeric	8	19[illegible]	5.46			Same as Run 44 with slight numerical modification.
46	Intersections and Unions	Triangular	2	22[illegible]	5.63	226	5.65	Same as Runs 44, 45. Complete failure.
55	Intersections and Unions	7-Feature Set	2	172	[illegible]7			Same as Runs 44, 45, 46. Complete failure.
58	Intersections and Unions	Triangular	[illegible]	85	[illegible]3	103	2.58	Same as Run 30 et seq. with size parameter 4.
60	Intersections and Unions	Triangular	9	76	[illegible].90	113	2.8[illegible]	Pattern set in original order.
61	Intersections and Unions	Triangular	9	81	2.02	106	2.65	First shuffle of pattern deck.
62	Intersections and Unions	Triangular	9	[illegible]8	2.2[illegible]	129	[illegible].23	Second shuffle of pattern deck.
63	Intersections and Unions	Triangular	9	[illegible]6	2.1[illegible]	114	2.85	Third shuffle of pattern deck.
64	Intersections and Unions	Triangular	[illegible]	81	[illegible].02	115	2.88	Fourth shuffle of pattern deck.
SRI 3	Intersections and Unions	7-Feature Set	[illegible]	36	[illegible]50			Determination of Composition matrix only.
		Triangular	[illegible]	76	[illegible].90	113	2.83	
		Alphanumeric	9	1[illegible]3	[illegible].23			
SRI 4, 5	Intersections and Unions Iterative	7-Feature Set	7	[illegible]7	0.7			Same as SRI 3 plus iteration of feature and composition matrix modification.
		Triangular	[illegible]	65	1.6	88	2.20	
		Alphanumeric	[illegible]	144	4.11			
47	Simple Iterative	Triangular	[illegible] ([illegible])	233	[illegible].83	268	6.70	Parameter ρ: 0.50 Number of independent features in
			[illegible] ([illegible])	23[illegible]	[illegible].78	260	6.50	0.65 parentheses. Parameter in
			9 ([illegible])	230	5.75	263	6.58	0.80 composition matrix only.

RE-ORDER No. [illegible]

Run Number	Type of Algorithm	Pattern Set	Number of Features	Train Set Errors Total	Train Set Errors Average	Test Set Errors Total	Test Set Errors Average			Comments
48	Simple Iterative	Triangular	9 (3)	233	5.83	267	6.68	Parameter ρ:	0.50	Same as Run 47 except starting with
			9 (6)	230	5.75	251	6.28		0.65	cluster features.
			9 (3)	230	5.75	263	6.58		0.80	
49	Simple Iterative	Triangular	9 (4)	233	5.83	268	6.70	Parameter ρ:	0.50	Parameter used in feature modification
			9 (3)	204	5.10	240	6.00		0.65	and composition matrix. Nilsson's
			9 (5)	202	5.05	241	6.03		0.80	features initially.
50	Simple Iterative	Triangular	9 (3)	233	5.83	267	6.68	Parameter ρ:	0.50	Same as Run 49 except starting with
			9 (4)	179	4.48	233	5.83		0.65	cluster features.
			9 (5)	178	4.45	215	5.38		0.80	
			9 (9)	100	2.50	120	3.00		0.95	
51	Sequential	7-Feature Set	14					Parameter θ:	1	Determined features only.
			14						2	
			14						3	
			10						4	
			6						5	
52	Sequential	Triangular	25					Parameter θ:	1	Determined features only.
			25						2	
			23						3	
			17						4	
			11						5	
53	Sequential	Triangular	25	31	0.78	44	1.10	Parameter θ:	2	Iterates modification of composition
			23	34	0.85	53	1.33		3	matrix.
			17	50	1.25	65	1.63		4	
			11	82	2.05	116	2.90		5	
			11	83	2.08	111	2.77		6	
54, 56	Sequential	7-Feature Set	14	0	0.00			Parameter θ:	1	Iterates modification of composition
			14	15	0.63				2	matrix.
			14	8	0.75				3	
			10	31	1.29				4	
			6	13	1.79				5	
57	Sequential	Alphanumeric	20	187	5.34			Parameter θ:	1	Iterates modification of composition
			20	266	7.60				2	matrix. Procedure terminated at 20
			20	158	4.5[illegible]				3	features.
			20	130	3.7[illegible]				4	
			20	122	3.49				5	
65, SRI 1, 2	Statistics	7-Feature Set Triangular Alphanumeric						Computes digram frequencies for each set of patterns.		

b. B5000 Runs - Checkout of Programming Techniques

Run 9 -- Checkout patterns. Demonstrated procedures for packing the patterns into words, reconstructing patterns from features, and printing out the patterns (one-line-per-pattern format).

Run 13 -- Checkout patterns. Demonstrated procedure for modification of the composition matrix.

Run 16 -- Checkout patterns. Full program utilizing both composition matrix and feature modification in training set, followed by composition matrix modification only for testing set. Status quo maintained if error not reduced by altering a composition-matrix element or feature element. Failed to produce perfect reconstitution. (N.B. Checkout patterns (14) made up of three vertical and three horizontal bars. Six features adequate for perfect reconstitution.)

Run 22 -- Checkout patterns. Same as Run 16 except that where the error was insensitive to a given matrix or feature element, that element was included on odd-numbered iterations. Produced perfect reconstruction of both training and testing patterns. (Eight training and six testing.)

Run 20 -- Checkout patterns. Same as Runs 16 and 22 except that where the error was insensitive to a given matrix or feature element, that element was included if previously omitted or omitted if previously included. Produced perfect reconstruction of both training and testing patterns.

c. B5000 Runs - Exhaustive Iterative Algorithm

Run 17 -- Triangular Patterns, Nilsson's initial features and composition matrix. Iterative modification of composition matrix and features for training set. Iterative modification of composition matrix only for testing set. If error insensitive to an element of compsition matrix or feature, that element was left unchanged.

9 Features

Training pattern errors, 126 bits total, 3.15 bits average
Testing pattern errors, 175 bits total, 4.38 bits average

Results using this rule are clearly inferior to those obtained for nine features in the 7090 runs.

Run 23 -- Triangular Patterns, etc., as in Run 17. Except that where the error was insensitive to a given element, that element was included on odd-numbered iterations and omitted on even-numbered iterations.

9 Features

Training pattern errors, 77 bits total, 1.93 bits average

Testing pattern errors, 116 bits total, 2.90 bits average

Results much better than given by "status quo" rule (#17) but still inferior to those obtained in the 7090 run.

Run 21 -- Triangular Patterns, etc., as in Runs 17 and 23. Except that where the error was insensitive to a given element, its value was changed. This is the same algorithm and the same rule used in the 7090 run.

9 Features

Training pattern errors, 66 bits total, 1.65 bits average

Testing pattern errors, 104 bits total, 2.60 bits average

Final results identical with those obtained on the 7090. Initial features averaged 7.22 elements per feature; final features averaged 6.0 elements per feature. Minimum error for training set achieved on 10th iteration.

Run time on B5000 was about 5 minutes, as compared with 30 minutes on the 7090.

During all subsequent runs, the rule used for modification of the composition matrix and feature elements was to change the values of those elements to which the error was insensitive.

Run 24 -- Triangular Patterns, Cluster feature input. Composition matrix initially empty. Initial features were three-element clusters, so that nine covered the retina with only two elements of overlap. Composition matrix and features modified iteratively for training set. Only composition matrix modified for testing set.

9 Features

Training pattern errors, 66 bits total, 1.65 bits average

Testing pattern errors, 86 bits total, 2.15 bits average

Total error on training set same as with Nilsson's initial features. Errors on testing set considerably smaller. Initial features were all three-element clusters; final features averaged 4.0 elements per feature -- 2/3 as large as when starting with Nilsson's features. Minimum error for training set achieved on 5th iteration -- half as many iterations as required with random initial features and composition matrix. There were only 81 bits total error for the training set on the first iteration, before feature modification.

N.B. Cluster features are somewhat matched to the triangular patterns, and their employment corresponds roughly to a reduction in resolution. Smaller features appear to be favorable with regard to minimizing errors in the testing set.

Run 59 -- Triangular Patterns, Cluster Feature input. Composition matrix initially empty. Features consisted of eight three-element clusters and one single element (center of 5 x 5 array) so that there was no overlapping. The feature set was not modified. Composition matrix was determined by the algorithm previously used for modification. Since the features do not overlap, iteration would serve no purpose.

9 Features

Errors in 1st 40 patterns, 76 bits total, 1.90 bits average

Errors in 2nd 40 patterns, 88 bits total, 2.20 bits average

The two sets of patterns were those identified as training and testing patterns elsewhere, but since no training took place in this run, use of those labels here would be misleading. Each pattern of each set is treated independently in the same way; hence, the lower error achieved for the 1st 40 patterns (training set) must be attributed to chance in the division of the arbitrarily ordered deck.

45

Lumping all 80 patterns together,

Errors in 80 patterns, 164 bits total, 2.05 bits average.

Pattern errors were distributed as follows:

8 patterns had 0 error

16 patterns had 1 error each

20 patterns had 2 errors each

36 patterns had 3 errors each

0 patterns had more than 3 errors.

The solid, triangular shapes of these patterns contain little structure; therefore, it is not surprising that simple reduction of resolution seems to be about as effective a bandwidth-compression technique as more sophisticated schemes for feature determination.

d. <u>B5000 Runs -- Intersections and Unions of Patterns</u>

<u>Run 25 -- Triangular Patterns</u>. This was an attempt to determine features by taking intersections and unions of patterns selected in a pseudo-random manner. At least one third of the patterns in the training set were involved in the determination of each of the nine features. If the current size of the evolving feature was smaller than a certain arbitrary parameter, union with the next selected pattern was performed. If larger than this parameter, the intersection was taken.

<u>9 Features</u>

Training pattern errors 158 bits total, 3.95 bits average

Testing pattern errors 191 bits total, 4.77 bits average

Five of the nine features resulting from this process were identical with five of the training patterns. The process obviously failed to distill the common features from the set.

<u>Run 27 -- Triangular Patterns</u>. In this attempt to determine features with a minimum of manipulation, the first nine training patterns were tentatively assigned as features. Each of the remaining training patterns was tested against the nine tentative features and combined with that feature with which it made the largest intersection. If the intersection was larger than an arbitrary parameter, the

intersection was taken; otherwise, the union was taken. Thus one feature was modified each time a pattern was considered, and the modification was performed before proceeding to consider the next pattern.

9 Features

Training pattern errors 98 bits total, 2.45 bits average

Testing pattern errors 128 bits total, 3.20 bits average

This procedure tends to modify the larger patterns selected as tentative features but to leave the smaller tentative features intact. Five of the resulting features were identical with the patterns used to initiate them. The other four started as large patterns and were very much modified.

Run 28 -- Triangular Patterns. Essentially the same as the algorithm of Run 27 except that the largest 9 patterns were selected as the tentative features. The remaining training patterns were then tested against each of the tentative features at its particular stage of evolution and combined with that one with which it had the maximum number of common elements. If this intersection was larger than the arbitrary size parameter then the intersection became the new tentative feature. If the intersection was equal to or smaller than the size parameter, then the new feature was the union of the old feature and the pattern being considered.

9 Features

Training pattern errors, 87 bits total, 2.18 bits average

Testing pattern errors, 119 bits total, 2.98 bits average

Starting with the largest patterns resulted in modification of all of the initial assignments; however, five of the nine resulting features were identical with patterns of the training set.

These results were an improvement over previous attempts at feature determination with a minimum of manipulation and led to a series of runs aimed at the empirical determination of some sort of optimization technique for the size parameter.

Runs 29, 30 -- Triangular Patterns. Parametric study to determine optimum value of size parameter for 9 features. Same algorithm as Run 28.

9 Features

Size Parameter (bits)	Training Pattern Errors (total bits -- average)		Testing Pattern Errors (total bits - average)	
3	109	2.73	113	2.83
4	76	1.90	113	2.83
5	79	1.98	106	2.65
6	87	2.18	119	2.98
7	87	2.18	119	2.98
8	99	2.48	128	3.20
9	103	2.58	139	3.48
12	112	2.80	154	3.85
15	140	3.50	176	4.40

The smallest error for the training set was achieved with the size parameter equal to 4. The average feature size for this case was 8.44 elements. Five of the features were identical with patterns of the training set.

The smallest average error for the combined pattern sets was achieved with the size parameter equal to 5. The average feature size for this case was 8.78 elements. Only one feature differed slightly from the set derived for the size parameter equal to 4.

As a starting guess, it was assumed that feature size, and hence any parameter relating to feature size, might well be a function of the ratio of retinal elements ($N = 25$ here) to number of features ($L = 9$). This ratio is here close to 3, so that $N/L + 1$ yields the size parameter which gave the best result for the training set.

Runs 31, 33 -- Triangular Patterns. Parametric study to determine optimum value of size parameter for 6 features. Same algorithm as Runs 28, 29, 30.

6 Features

Size Parameter (bits)	Training Pattern Errors (total bits -- average)		Testing Pattern Errors (total bits - average)	
2	132	3.30	168	4.20
3	144	3.60	179	4.48
4	118	2.95	131	3.27
5	126	3.15	126	3.15
6	111	2.77	130	3.25
7	123	3.08	138	3.45
8	126	3.15	157	3.93
9	119	2.98	139	3.48
10	119	2.98	140	3.50
11	122	3.05	152	3.80
12	122	3.05	152	3.80

Smallest error for training set and for the combined pattern sets was achieved for a value of 6 for the size parameter. Average feature size was 9.00 elements. Only one feature was identical with an input pattern.

In this case, N/L is about 4, so that the expression N/L + 2 yields the size parameter which gave the best result.

Runs 32, 34 -- Triangular Patterns. Parametric study to determine optimum value of size parameter for 12 features. Same algorithm as Runs 28, 29, 30, 31, 33.

12 Features

Size Parameter (bits)	Training Pattern Errors (total bits -- average)		Testing Pattern Errors (total bits - average)	
2	81	2.02	105	2.63
3	74	1.85	90	2.25
4	78	1.95	101	2.52
5	78	1.95	101	2.52
6	76	1.90	117	2.93
7	76	1.90	117	2.93
8	106	2.65	137	3.43
9	98	2.45	136	3.40
10	96	2.40	144	3.60
11	96	2.40	144	3.60
12	96	2.40	144	3.60

Smallest error for training set and for the combined pattern sets was achieved with a value of 3 for the size parameter. Average feature size was 6.00 elements. Ten of the twelve features were identical with input patterns.

In this case, N/L is about 2, so that the expression N/L + 1 yields the size parameter which gave the best results.

For 6, 9, and 12 features, the product of the optimum size parameter and the number of features is $36 = (5 + 1)^2$, where square 5 X 5 patterns are involved. This might generalize to an expression of the form $(X + 1) (Y + 1)/L$ for the size parameter, where X and Y are the linear dimensions of the retina, but experiments with various sizes and shapes of arrays would be required to substantiate it.

Run 35--Triangular Patterns. Using the same algorithm as applied above, nine initial features were determined, with the size parameter equal to 4. These features were then modified in the iterative procedure used in the 7090 runs and on B5000 Run 21. Both features and composition matrix were so modified for the training set; only the composition matrix being evaluated and modified for the testing set.

9 Features

Training pattern errors, 65 bits total, 1.63 bits average
Testing pattern errors, 88 bits total, 2.20 bits average

Comparison with results of Run 21 indicates that the initial features as determined here serve as a better starting point for modification than Nilsson's randomly-generated initial features. Although the improvement in the final result is only one bit in 40 patterns for the training set, there is a 16 bit improvement for the 40-pattern testing set. The final features obtained in this run average 4.89 elements per features, as against 6.0 elements per feature when starting with the Nilsson features.

The final result here is very close to that obtained in Run 24, which started with the cluster-type features. In that case, the total errors were 66 and 86 bits for the training and testing sets, respectively.

Run 36--Triangular Patterns. The feature-determining algorithm selects the largest patterns as tentative features and then compares each of the remaining patterns with the feature set, combining each with the feature with which it has the largest intersection. Each pattern is utilized once in this process, either as an initial tentative feature or combined by intersection or union with one of the tentative features. Run 36 made provision for repeating the second part of this algorithm a number of times, i.e., again comparing each pattern of the training set with each of the features and combining it with one according to the same rules. Each such revision utilized each pattern an additional time.

9 Features

Revisions	Training Pattern Errors (total bits -- average)		Testing Pattern Errors (total bits - average)	
0	76	1.90	113	2.83
1	83	2.08	102	2.55
2	85	2.13	101	2.52
3	85	2.13	101	2.52
4	85	2.13	101	2.52
5	85	2.13	101	2.52
6	85	2.13	101	2.52

Clearly, revision did not improve the error for the training set. Revision led to slight reduction in the average size of the features, from 8.44 elements without revision to 7.44 elements after the first revision and to 7.22 elements after the second and subsequent revisions. The features were not altered by the revision process after the second revision.

Run 37 -- Alphanumeric Patterns. The algorithm initiated in Run 28 and investigated in subsequent runs on the Triangular Patterns (largest L patterns as tentative features, modified by intersections or unions with the remaining patterns) was run here with 35 Alphanumeric Patterns taken from the printer on the IBM card punch. These are 5 by 7 arrays. The size parameter determining whether the intersection or the union with a given tentative feature was taken was evaluated by the formula N DIV L + 1, which produced a value of 5 with 35 retinal elements and 8 features, since DIV produces the unrounded division of integers. The first 20 letters of the alphabet (A through T) served as the training set; the 15 remaining letters and numerals (U through Z and 1 through 9) served as the testing set. One set of results was obtained by iterative modification of the composition matrix only for both sets.

8 Features

Training pattern errors, 75 bits total, 3.75 bits average

Testing pattern errors, 122 bits total, 7.47 bits average

Subsequently, both the composition matrix and the features were modified through 10 iterations for the training set. Only the composition matrix was modified for the testing set.

8 Features

Training pattern errors, 58 bits total, 2.90 bits average

Testing pattern errors, 117 bits total, 7.80 bits average

Since the numerals exhibit rather different characteristic shapes from the capital letters, the poor performance on the testing set is not surprising. Considerable error improvement with modification of the features indicates that the algorithm determining the initial features is not working too well. It was reasoned that starting with the largest patterns might not result in sufficient diversity among the tentative features.

Run 39 -- Alphanumeric Patterns. The algorithm used above was modified in the following way. Each time a pattern was chosen as a tentative feature, all the rest of the patterns were tested against it. Those patterns which intersected the chosen tentative feature in a number of elements equal to or greater than twice the size parameter (2 x 5 = 10 elements, for this case) were eliminated from consideration as possible tentative features, regardless of their sizes. Otherwise, the procedure was the same as that above. Additional printout was provided during the feature determining process so that the detailed steps followed could be reconstructed by hand and analyzed. All 35 patterns were used as a training set. After determination and iterative modification of the composition matrix only, the results achieved were:

8 Features

Training pattern errors, 186 bits total, 5.31 bits average

After 10 iterations involving modification of the features as well as the composition matrix, this was reduced to

8 Features

Training pattern errors, 136 bits total, 3.89 bits average

These results cannot be easily compared with the previous run because the size of the training set was different. The fact that feature modification led to significantly improved results indicates that the algorithm determining the initial features should be capable of improvement. Many patterns were removed from consideration early in the feature determination because of large intersections with the first couple of tentative features.

Run 40 -- Alphanumeric Patterns. Only one factor was altered from the above run. This time, the criterion for elimination of a pattern from further consideration as a tentative feature was that its intersection with the last-chosen tentative feature be equal to or greater than three times the size parameter (3 x 5 = 15, for this case). With modification of the composition matrix only, the following errors were obtained.

8 Features

Training pattern errors, 217 bits total, 6.20 bits average

After feature and composition-matrix modification:

8 Features

Training pattern errors, 152 bits total, 4.34 bits average

This modification of the algorithm produced poorer results, both before and after feature modification.

Run 41 -- Alphanumeric Patterns. It was thought that perhaps a better test as to whether a pattern should be eliminated from consideration as a tentative feature would be one based on what fraction of that pattern overlapped the previously chosen tentative feature. Thus the criterion was changed so that if the overlap were equal to or greater than 3/4 of the elements in the pattern under consideration, then it was eliminated. With modification of the composition matrix only, the result achieved was:

<u>8 Features</u>

Training pattern errors, 167 bits total, 4.77 bits average

After iterative modification of both the features and the composition matrix, the result was:

<u>8 Features</u>

Training pattern errors, 155 bits total, 4.43 bits average

This is the best result obtained, to this point, before feature modification. The result after modification is still not as low as the minimum achieved in Run 39; however, the chief objective at this point was to obtain the best possible result without such modification, since iteration of feature modification is expensive in terms of computer time.

It was noted in this run that when the 8th tentative feature had been chosen, all of the training patterns had either been used or eliminated from consideration. Thus it was seen that this algorithm could be slightly modified to determine the number of features, as well as their configurations.

<u>Run 43 -- Alphanumeric Patterns</u>. In order to eliminate one arbitrary parameter, namely, the number of features to be determined, the program was altered so that when each of the patterns had either been used as a tenetative feature or eliminated from consideration, the number of features was frozen. This run produced eight features and error counts identical with those obtained in Run 41.

<u>Run 44 -- Alphanumeric Patterns</u>. In order to make the feature determination process independent of the order in which patterns were loaded into the system, the process was altered to order the patterns by size. To check the sensitivity of the algorithm to the parameter determining whether a pattern should be retained as a possible feature or eliminated, rounded division was used when the number of elements in

each pattern was multiplied by 3/4. (Unrounded division of integers had been previously employed.) This algorithm found 10 features and produced the following results:

10 Features

Training pattern errors 194 bits total, 5.54 bits average

Thus the errors were greater than those produced in Runs 41 and 43, even though there were two additional features.

Run 45 - Alphanumeric Patterns. Patterns were ordered again, as in Run 44, with the largest patterns considered first. Unrounded division was used when multiplying pattern size by 3/4. Again, eight features were found, as in Runs 41 and 43, where tentative features were determined by considering the patterns in order of decreasing size. In performing unions and intersections, however, the remaining patterns are now also considered in size sequence, leading to different final features. The results were:

8 Features

Training pattern errors, 191 bits total, 5.46 bits average

Thus ordering of the patterns did not improve upon the results obtained in Runs 41 and 43.

Run 46 -- Triangular Patterns. Although the algorithm employed in Run 45 was not optimized for the alphanumeric patterns, it had evolved from experiments in which only the alphanumerics were used (Runs 37 through 45), and it was anticipated that its performance on a pattern set with very different statistics would be poor. This was tested by running the Triangular Patterns with this algorithm, letting the feature-determining process also determine the number of features. Performance was as follows:

2 Features

Training pattern errors, 225 bits total, 5.63 bits average
Testing pattern errors, 226 bits total, 5.65 bits average

By the time two tentative features had been found, all the rest of the 40 training patterns had been eliminated, since all of the smaller patterns are essentially completely contained in the larger ones, as can be seen from the two patterns selected as tentative features:

```
x x x x x          x
x x x x x        x x x
x x x x x      x x x x x
  x x x        x x x x x
    x          x x x x x
```

After taking intersections and unions with the remaining 38 training patterns, the features had evolved to:

```
  x x x x      x x x x x
x x x x x      x x x x x
  x x x x      x x x x x
    x x        x x x x x
               x x x x x
```

Obviously, many of the "reconstructed" patterns were either wholly empty or wholly filled. The remaining ones were represented by the first "feature." Error rate, expressed in total bits or bits per pattern, is inadequate as a measure of such complete failure.

* * *

Chronologically, experiments were next run on two quite different algorithms, reported in a later section (Runs 47 through 57). A new pattern set was introduced in testing one of them, a set of 24 patterns made up entirely of seven strokes on the 5-by-5 retina as shown in Fig. 14 of Ref. 1. Since this was the first set of any size which we had used that was guaranteed reconstructable in terms of a small number of features, it was of interest to run this set with the "intersections and unions" algorithm.

<u>Run 55--Seven-feature Patterns</u>. Same algorithm as Runs 45 and 46, which identified only two "features."

<u>**2 Features**</u>

Training pattern errors 172 bits total, 7.17 bits average

Again, the result was total failure.

* * *

Looking over the results of runs to date, it was decided that the modifications made to this algorithm in Runs 37 through 45 had been unprofitable, as far as obtaining a general algorithm. The results obtained during the parametric study, Runs 29 through 34, could no longer be duplicated, to say nothing of improved. One question remained, however, with regard to the generality of the earlier form of the algorithm: Since it processed patterns in the (arbitrary) order in which they were introduced, how sensitive might the result be to the order of the patterns? The next several runs sought to answer this question.

Runs 58, 60, 61, 62, 63, 64 -- Triangular Patterns. The same algorithm as that used in Runs 30 through 34, except that the size parameter was set equal to one plus the rounded quotient of the number of retinal points divided by the number of features sought, in these cases, 1 + 25/9 = 4. All runs were identical except that the training pattern deck was shuffled between runs:

9 Features

Training pattern errors (total bits -- average)		Testing pattern errors (total bits - average)		
85	2.13	103	2.58	Note 1
76	1.90	113	2.83	Note 2
81	2.02	106	2.65	
88	2.20	129	3.23	
86	2.15	114	2.85	
81	2.02	115	2.88	

Note 1: Patterns in original order, but slight change in reprogramming this algorithm had the effect of comparing the features with each pattern in reverse order.

Note 2: Patterns in original order, algorithm corrected. Result same as in the first experiment of Run 30.

Mean values for the six runs:

Training pattern errors 82.83 bits total, 2.07 bits average

Testing pattern errors 113.33 bits total, 2.83 bits average

Root-mean-square deviation from the mean for the six runs:

Training patterns: 3.98 bits total, 0.10 bits average

Testing patterns: 8.44 bits total, 0.21 bits average

The spread was really quite small, although the original run proved to yield the lowest error for the training set. It was concluded that this algorithm is not particularly sensitive to the order of presentation of the Triangular Pattern set.

Run SRI 3 -- All three pattern sets. Feature determination plus modification of the composition matrix only, as in the above runs. Run on the SRI B5000.

Seven-Feature Patterns

Training pattern errors 36 bits total, 1.50 bits average

Although this is not a high error figure, it should be zero, since the patterns are known to be capable of reconstruction from seven features.

Triangular Patterns -- 9 Features

Training pattern errors 76 bits total, 1.90 bits average
Testing pattern errors 113 bits total, 2.83 bits average

Same result as previously obtained -- same algorithm.

Alphanumeric Patterns -- 9 Features

Training pattern errors 183 bits total, 5.23 bits average

This is a larger error than had been obtained with 8 features with some of the modifications of this algorithm that turned out to be fairly specific to this pattern set.

Runs SRI 4, 5 -- All three pattern sets. Feature determination followed by iterative modification of both composition matrix and features. Run SRI 4 employed ten iterations in the training process. Since some values appeared to be still changing at this point, Run SRI 5 employed twenty iterations for training. Final results of the two runs were identical, as far as the error summary is concerned, but some of the features differed in the two sets and, hence, some of the reconstructions also differed.

Seven-Feature Patterns

Training pattern errors 17 bits total, 0.71 bits average

Even with iterative modification of the features, the error was not reduced to its known minimum value of zero.

Triangular Patterns -- 9 Features

Training pattern errors 65 bits total, 1.63 bits average
Testing pattern errors 88 bits total, 2.20 bits average

This result duplicates Run 35.

Alphanumeric Patterns -- 9 Features

Training pattern errors 144 bits total, 4.11 bits average

This result is comparable to, but not as low as, the result obtained in Run 39 with 8 features.

Although the iterative algorithm, involving modification of both the composition matrix and the features, has produced the lowest error rates and has been used as the standard of comparison, it is illustrated again here that it does not guarantee the minimum error achievable with a given number of features.

e. B5000 Runs -- Simplified Iterative Algorithm

The following experiments were suggested by Nils Nilsson in an attempt to provide a simplified scheme for iterative modification of features that would be faster than the exhaustive algorithm run on the 7090 and subsequently on the B5000.

Run 47 -- Triangular Patterns, Nilsson's initial features, composition matrix initially empty. In this algorithm, a feature was included in the reconstruction of a pattern if more than a specified fraction, ρ, of the feature was included in the pattern. An element was included in a feature if that element was present in more than half the patterns to which it was assigned for reconstruction.

There is no interaction among features in this algorithm, and the computation is simple and fast. Three values of were used in this run:

9 Features

Parameter ρ	Training Pattern Errors (total bits -- average)		Testing Pattern Errors (total bits - average)	
0.50	233	5.83	268	6.70
0.65	231	5.78	260	6.50
0.80	230	5.75	263	6.58

Independence of the features resulted in duplication of features. In the first and last cases, only four distinct features resulted. The other case produced six distinct features. Errors did not always decrease with successive iterations.

Run 48 -- Triangular Patterns, Cluster features, composition matrix initially empty. Aside from the fact that the initial features employed were the cluster set (see Run 24), this run was precisely the same as Run 47:

9 Features

Parameter ρ	Training Pattern Errors (total bits -- average)		Testing Pattern Errors (total bits - average)	
0.50	233	5.83	267	6.68
0.65	230	5.75	251	6.28
0.80	230	5.75	263	6.58

Starting with the cluster features demonstrated that this algorithm can be highly divergent. The initial feature assignment in each case produced a total error of 81 bits, or just over two bits per pattern for the training set. Iteration of the assignment and feature-modification processes nearly tripled the error. Only three distinct features were produced in the first and last cases; six, in the other.

Run 49 -- Triangular Patterns, Nilsson's initial features, composition matrix initially empty. In this run, the algorithm was modified to introduce the same parameter, ρ , used in determining the compsition matrix into the feature-modification procedure. An element was included in a feature if that element was present in more than the fraction ρ of the patterns to which it was assigned for reconstruction. The results, for the same range of parameter values, were:

9 Features

Parameter ρ	Training Pattern Errors (total bits -- average)		Testing Pattern Errors (total bits - average)	
0.50	233	5.83	268	6.70
0.65	204	5.10	240	6.00
0.80	202	5.05	241	6.03

The first case is, of course, the same as that of Run 47, producing four distinct features. The others appear somewhat better. Only three distinct features were produced in the second case, and five in the third. Starting with these initial features, this modification of the algorithm produced some convergence toward minimum errors.

Run 50 -- Triangular Patterns, Cluster features, composition matrix initially empty. Aside from the fact that the initial features employed were the cluster set and that one additional case was added (ρ = 0.95), this run was precisely the same as Run 49:

9 Features

Parameter ρ	Training Pattern Errors (Total bits -- average)		Testing Pattern Errors (total bits - average)	
0.50	233	5.83	267	6.68
0.65	179	4.48	233	5.83
0.80	178	4.45	215	5.38
0.95	100	2.50	120	3.00

The first case duplicates the first case of Run 48, producing only three distinct features. Total error for the training set diverged from 81 to 233 in 10 iterations. The second case produced four distinct features but diverged from 81 to 179 bits total error. The third case produced five distinct features and diverged from 81 to 178 bits total error. In the final case, all nine features were distinct after 10 iterations, and the error had changed from 114 to 100 bits total for the training set. Seven of the nine original features emerged unchanged after 10 iterations. Two bits each had been added to the other two features.

The final case exhibited somewhat strange behavior, illustrating the instability of this algorithm. When the number of patterns or number of elements in a pattern was multiplied by ρ, the result was rounded and stored as an integer; thus in the comparisons which followed, equality was always possible. The rule for handling equality which had proved most useful in the earlier iterative algorithm (Run 21, at seq.) was again used here, namely, reversal of the decision made on the previous iteration. In an algorithm in which the several features interacted, this rule provided a "dither" which tended to move the error off of any plateau encountered during convergence toward a minimum value. Occasionally, a cyclic behavior was observed. Here, without interaction among the features, cyclic behavior appears to be highly probable. In the third case, pattern errors were cyclic for 27 of the 40 training patterns and constant after the first iteration for the other 13. The total error fluctuated between two values. After initial assignment of features to patterns (features as yet unmodified), the error was 114 bits total. An iteration consisted of feature modification, reassignment of features to patterns for reconstruction, and error determination. After Iterations 1, 2, 4, 5, 7, and 8, the total error for the training set was 387 bits. After Iterations 3, 6, and 9, the total error was 100 bits. A final feature modification, completing ten total cycles through the process, left the result unchanged at 100 bits. Clearly, although the final error was less than the initial error, the process could not be said to "converge."

f. B5000 Runs -- Sequential Algorithm

Run 51 -- Seven-Feature Patterns. The pattern set employed here consists of 24 patterns on a 5 x 5 retina, as drawn in Fig. 14 of Reference 1. All of the patterns can be perfectly reconstructed from a set of seven simple features (three horizontal and three vertical strokes, and one diagonal). This run merely determined features for a range of values of the size parameter θ. Pattern reconstruction and error determinations were not made.

62

Parameter θ	Number of Features
1	14
2	14
3	14
4	10
5	6

The 14 features determined in the first case included all of the seven significant features generated by the "Parallel Algorithm" and displayed in Fig. 15 of Reference 1. The other "features" consisted of one or two elements, totally redundant. None of the other cases produced a feature set capable of perfect reconstruction of the input patterns.

Run 52 -- Triangular Patterns. Determination of features only:

Parameter θ	Number of Features
1	25
2	25
3	23
4	17
5	11

An upper limit on the number of features to be sought was entered into these runs, along with the other data. In this case, the upper limit was set at 25, equal to the number of retinal elements in the patterns under consideration. The first case yielded one-element features, each a separate retinal element. Even when the minimum feature size was set at five elements (final case), eleven features were determined.

Run 53 -- Triangular Patterns. After determining features, the procedure assigned them to patterns in which they were totally included, as a first step in reconstruction. Following this, the procedure for modification of the composition matrix was executed through five iterations. Only the training set was used for feature determination.

Parameter θ	Number of Features	Training Pattern Errors (total bits -- average)		Testing Pattern Errors (total bits - average)	
2	25	31	0.78	44	1.10
3	23	34	0.85	53	1.33
4	17	50	1.25	65	1.63
5	11	82	2.05	116	2.90
6	11	83	2.08	111	2.77

In the first case, although there were 25 "features," equal to the number of retinal elements, the training pattern set was not perfectly reconstructed. At the other end of the experiment, the final case allows a minimum feature size of six elements, yet 11 features are found and the errors are not better than those produced by other algorithms for nine features.

Run 54 -- Seven-feature patterns. Same as Run 53, except that the input patterns were the set derived from seven features. All 24 patterns were used as a training set; the testing set being empty.

Parameter θ	Number of Features	Training Pattern Errors (total bits -- average)	
1	14	0	0.00
2	14	15	0.63
3	14	18	0.75
4	10	31	1.29
5	6	43	1.79

Only in the first case did this algorithm produce features sufficient for perfect reproduction of these patterns synthesized from seven features, and then it produced an equal number of redundant "features." As the size parameter was increased, to reduce the number of features, not only the redundant members of the set were eliminated, but some of the essential members as well.

Run 56 -- Seven-Feature patterns. Same as Run 54 except that a new output format was employed. For efficiency in earlier runs, patterns had been printed out in a one-line format, as

XXXXX X.... XXXX. X.... X....

To one working closely with these patterns, this is easy to interpret; however, to facilitate discussion with others not so intimately involved, the more graphic rectangular array output format was programmed:

```
XXXXX    ....X    X...X
X....    ....X    X..XX
XXXX.    ....X    X.X.X    Etc.
X....    ....X    XX..X
X....    XXXXX    X...X
```

Ten 5-by-5 or 5-by-7 patterns print out simultaneously in this format. Where the patterns have some meaning other than as arbitrary arrays of elements, such as the alphanumeric set, this format permits immediate subjective evaluation as to whether this meaning has been preserved in the reconstruction of a given pattern.

Summary results of this run were identical with those obtained in Run 54.

Run 57 -- Alphanumeric Patterns. This set of patterns was run through the Sequential Algorithm to test its performance on a set of patterns not constructed from a limited set of features. In order to avoid excessive run time, computation was terminated after the determination of 20 features for each case.

Parameter θ	Number of Features	Training Pattern Errors (total bits	-- average)
1	20	187	5.34
2	20	266	7.60
3	20	158	4.51
4	20	130	3.71
5	20	122	3.49

It might have been interesting to continue this experiment with higher values of the size parameter; however, it was reasoned at the time that failure to converge to fewer than 20 features with a minimum size of five elements amounted to failure of the algorithm to perform satisfactorily with this pattern set.

g. B5000 runs -- Digram Statistics of Pattern Sets

Runs 65, SRI 1, SRI 2 -- All three pattern sets. These runs were not feature detection algorithms, but merely computations of the frequencies of element pairs in the three sets of patterns. These digram frequencies are presented in triangular arrays. The diagonal elements are the numbers of times each element occurred with itself in the pattern set, and are thus the frequencies of the individual elements.

The line patterns, Seven-Feature set and Alphanumeric set, exhibit wide variations in the digram frequencies. The "blob" patterns, i.e., the Triangular set, exhibit small variations, with a fairly uniform decrease in frequency with distance between elements.

REFERENCES

1. H. D. Block, N. J. Nilsson, and R. O. Duda, "The Determination and Detection of Features in Patterns," Proceedings of the 1963 COINS Symposium (to be published).

2. C. E. Shannon and W. Weaver, The Mathematical Theory of Communication, (University of Illinois Press, Urbana, Illinois, 1949).

3. D. A. Huffman, "A Method for the Construction of Minimum Redundancy Codes," Proc. IRE, vol. 40, no. 10, pp. 1098-1101 (September 1952)

4. N. Abramson, Information Theory and Coding, (McGraw-Hill Book Co., Inc., New York, 1963).

5. N. J. Nilsson, "Some Experiments in Reconstituting Images from Features," Interim Progress Report 1, Prepared for Jet Propulsion Laboratory, Sub-contract 950687 under NAS7-100, SRI Project 4621, Stanford Research Institute, Menlo Park, California (October 1963).

6. W. E. Schreiber and C. F. Knapp, "TV Bandwidth Reduction by Digital Coding," 1958 IRE National Convention Record, Part IV, p. 88.